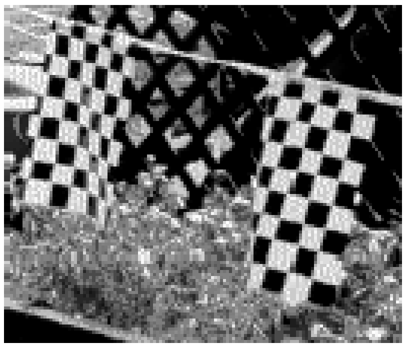
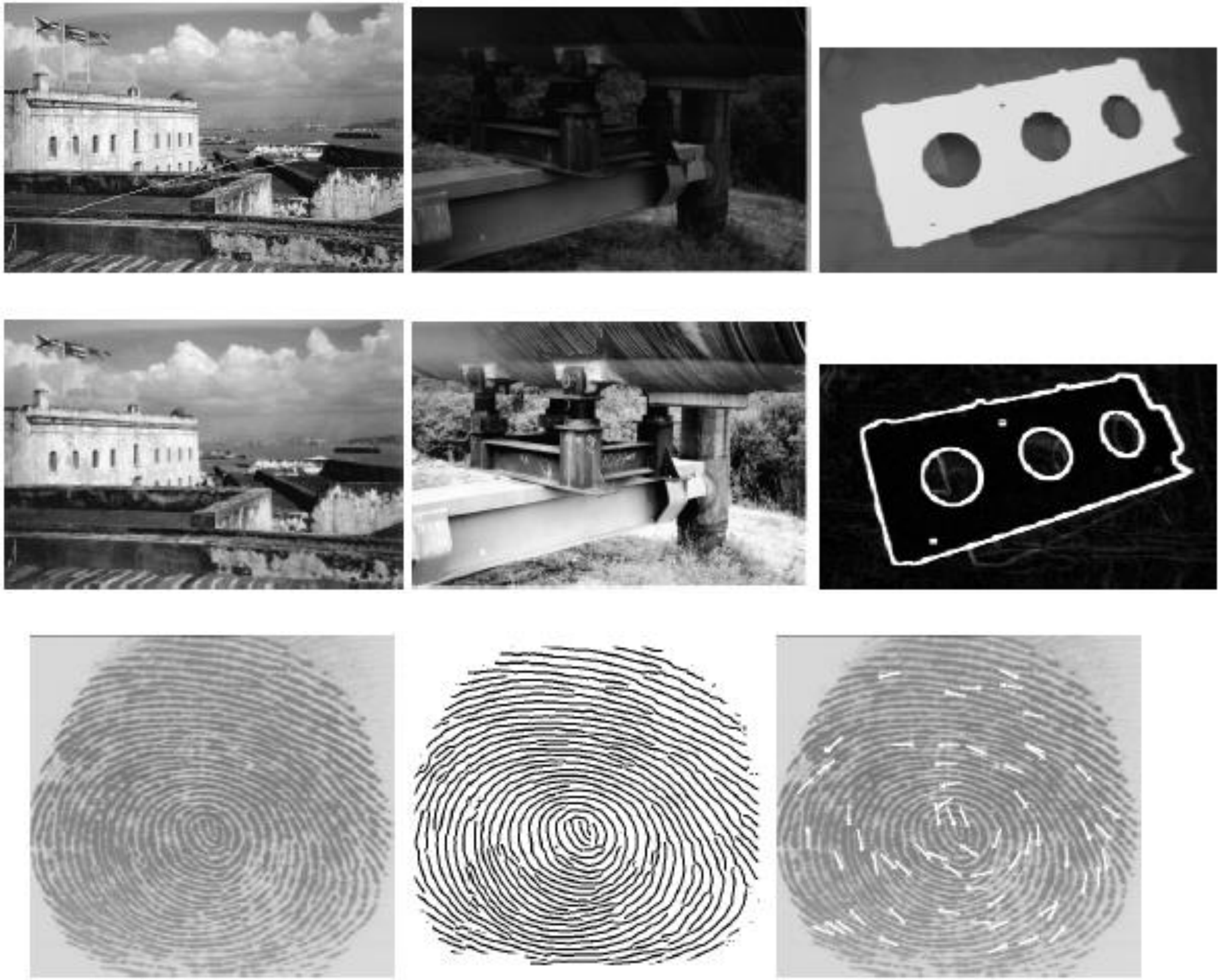
5º CAPÍTULO

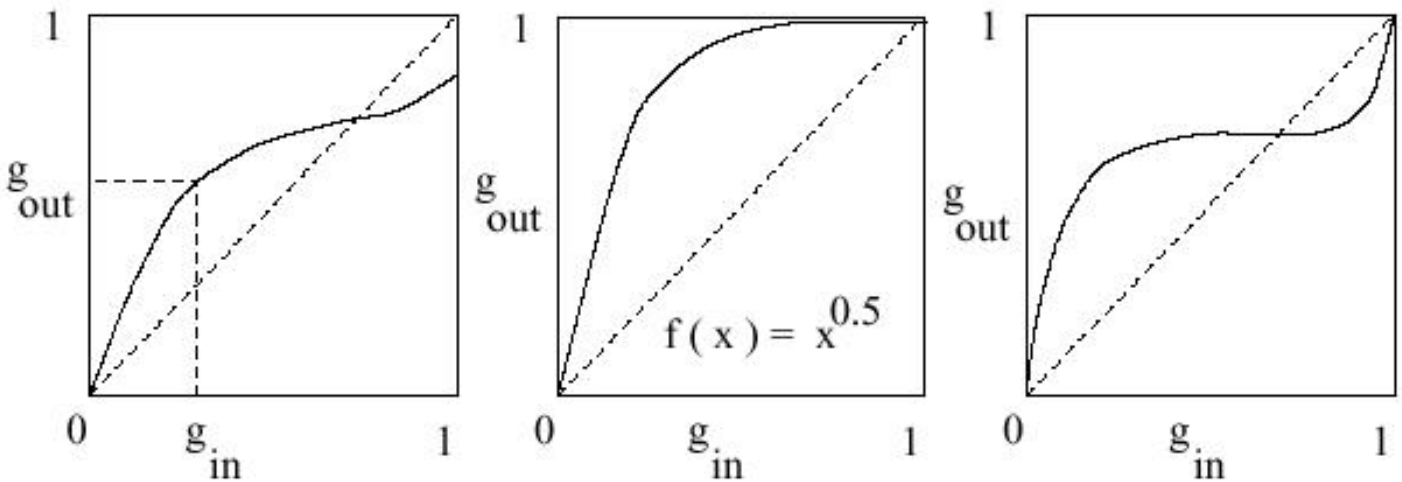
Pré-Processamento de Imagem

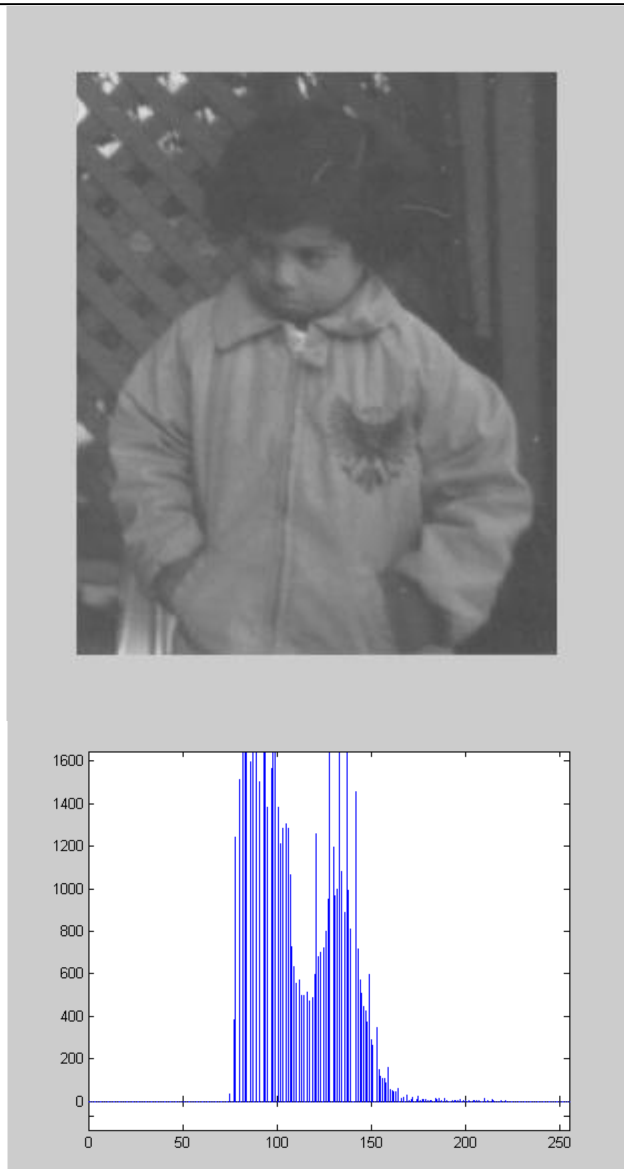
Prof. Arnaldo Abrantes / anotado por Prof. Nuno Pinho da Silva

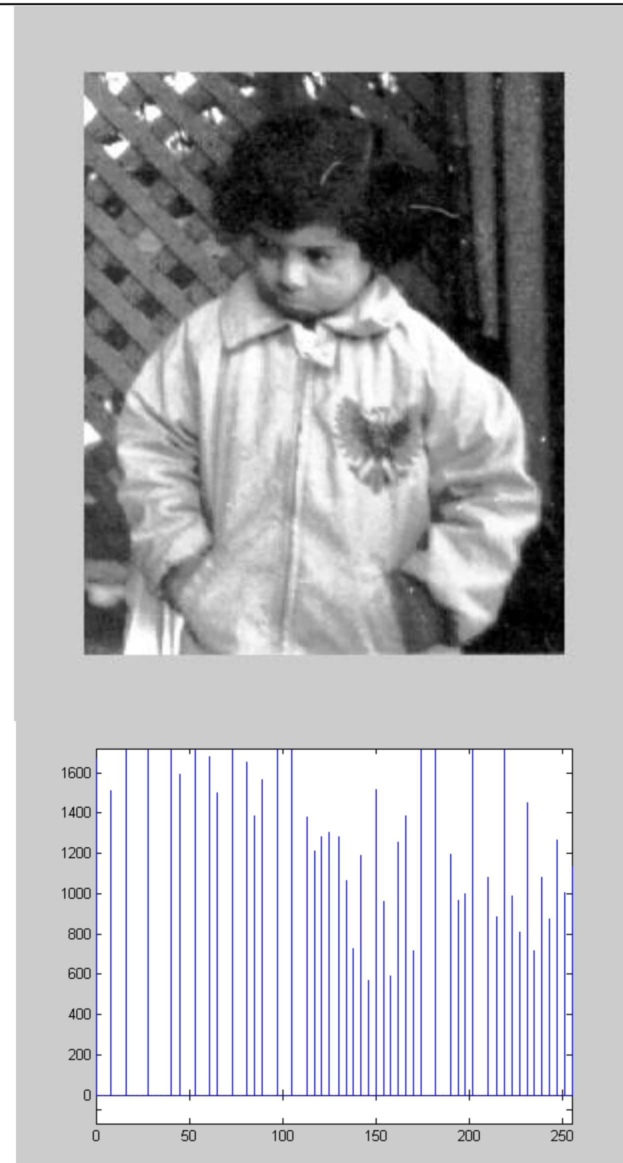
Necessidade de pré-processamento





- Correcção gama $\longrightarrow$ $f(x) = O_{\min} + \dfrac{O_{\max} - O_{\min}}{\left(I_{\max} - I_{\min}\right)^{\gamma}}\left(x - I_{\min}\right)^{\gamma}$


1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250


1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250

Remover pequenas regiões

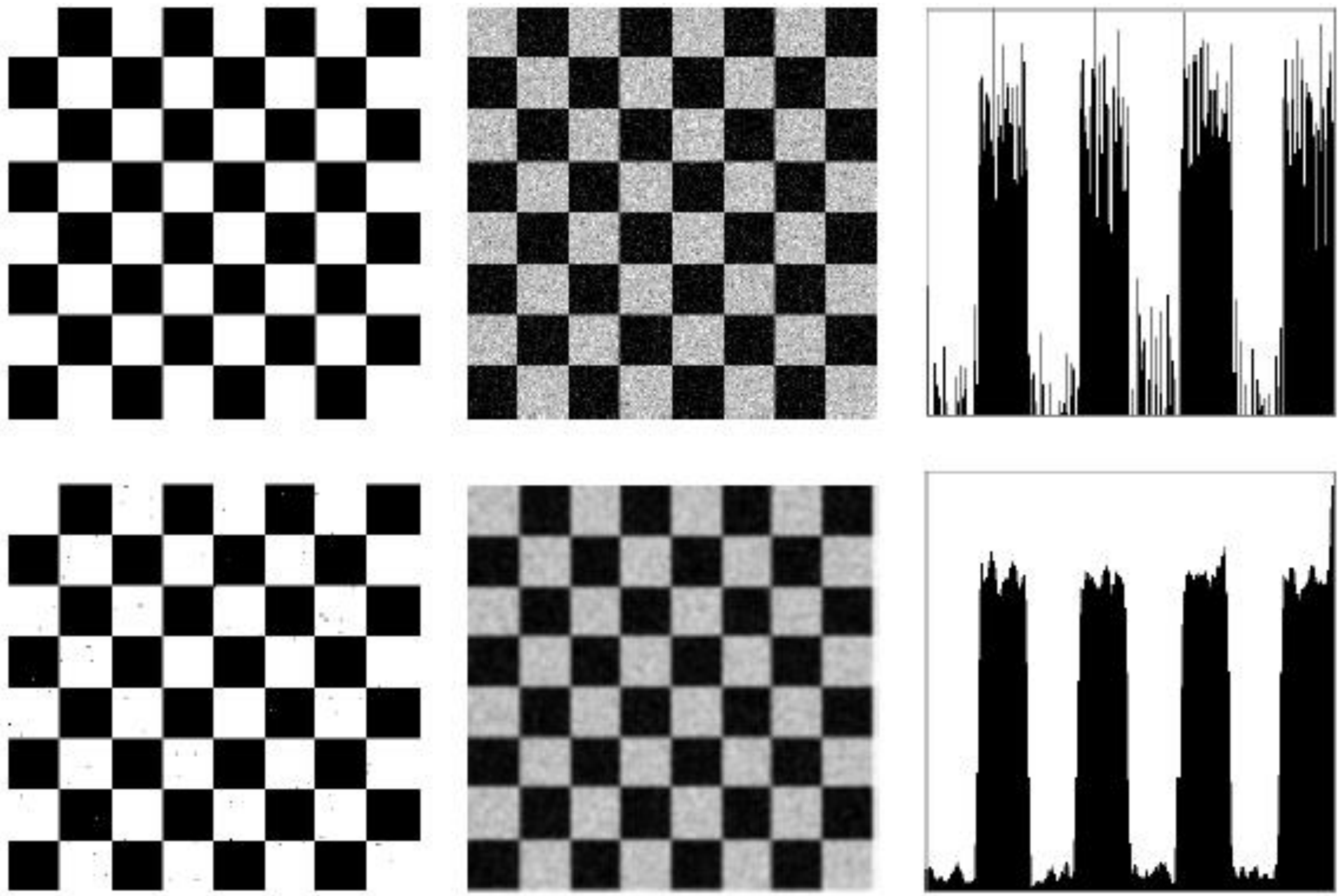
← Remoção de ruído salt & pepper

$$\begin{array}{|c|c|c|}\hline 1&1&1\\\hline 1&0&1\\\hline 1&1&1\\\hline\end{array} \Rightarrow \begin{array}{|c|c|c|}\hline 1&1&1\\\hline 1&1&1\\\hline 1&1&1\\\hline\end{array} \quad ; \quad \begin{array}{|c|c|c|}\hline 0&0&0\\\hline 0&1&0\\\hline 0&0&0\\\hline\end{array} \Rightarrow \begin{array}{|c|c|c|}\hline 0&0&0\\\hline 0&0&0\\\hline 0&0&0\\\hline\end{array}$$

$$\begin{array}{|c|c|c|}\hline X&X&X\\\hline X&L&X\\\hline X&X&X\\\hline\end{array} \Rightarrow \begin{array}{|c|c|c|}\hline X&X&X\\\hline X&X&X\\\hline X&X&X\\\hline\end{array} \quad ; \quad \begin{array}{|c|c|c|}\hline &X&\\\hline X&L&X\\\hline &X&\\\hline\end{array} \Rightarrow \begin{array}{|c|c|c|}\hline &X&\\\hline X&X&X\\\hline &X&\\\hline\end{array}$$

Remoção de componentes conexos cuja área é pequena →

Necessidade de operação de suavização



- Suavização (filtragem passa-baixo) de imagem

  - filtro de média (*box filter*)

$$O(r,c)=\left(\sum_{i=-N}^{N}\sum_{j=-N}^{N} I(r+i,c+j)\right)\Big/(2N+1)^2$$

  - filtro gaussiano

$$O(r,c)=\sum_{i=-N}^{N}\sum_{j=-N}^{N} g(i,j)I(r+i,c+j)$$

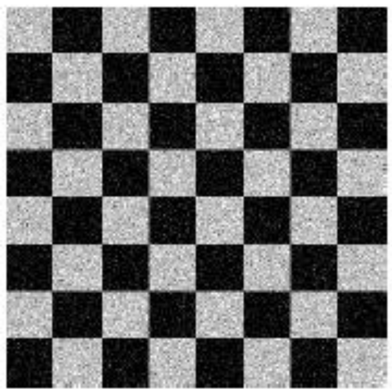
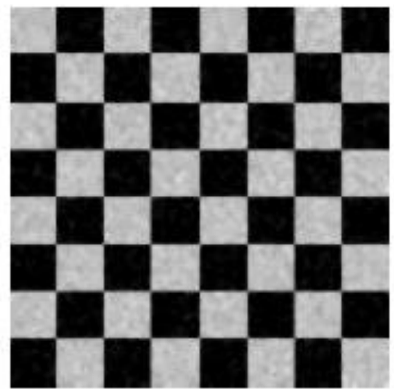
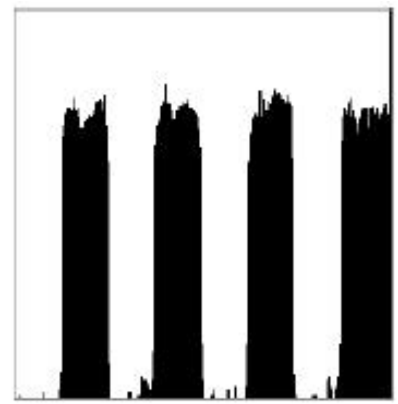
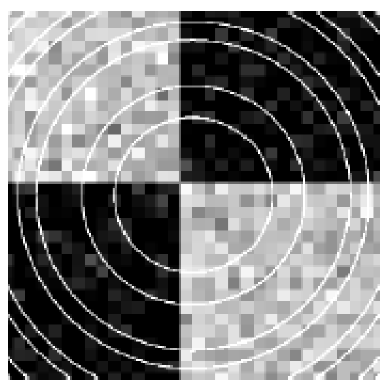
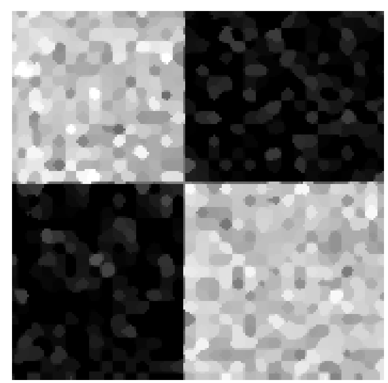
$$g(x,y)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{d^2}{2\sigma^2}}$$

$$d=\sqrt{(x-x_c)^2+(y-y_c)^2}$$

Filtragem de mediana

- Seja $\mathbf{A}[i]_{i=0,\cdots,n-1}$ uma lista ordenada de números reais. A ***mediana*** do conjunto **A** é o valor **A**[(n-1)/2]

  – Exemplos

Filtragem temporal com filtro de mediana

Filtragem de Mediana da Sequência

Compute output image pixel G[r,c] from neighbors of input image pixel F[r,c].

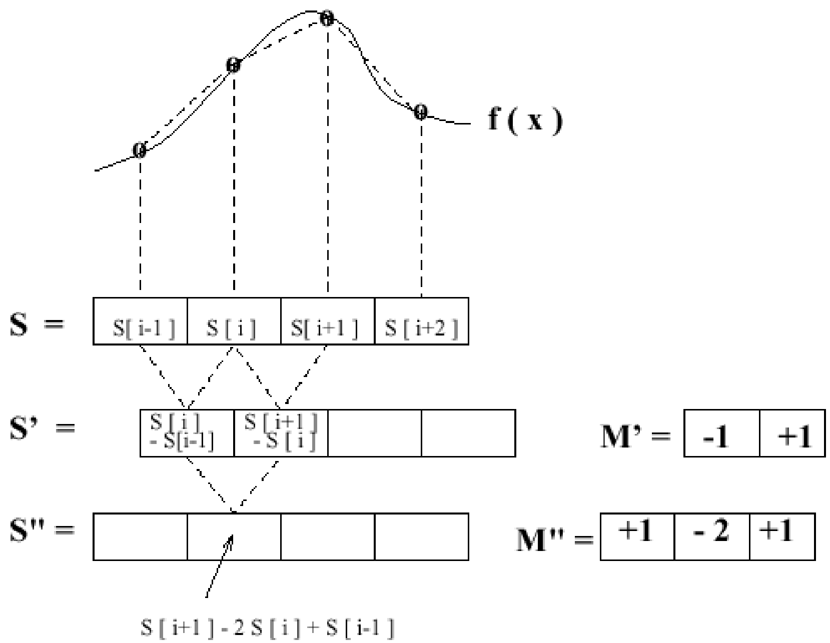
F[**r**, **c**] is an input image of MaxRow rows and MaxCol columns;
F is unchanged by the algorithm.
G[**r**, **c**] is the output image of MaxRow rows and MaxCol columns.
The border of **G** are all those pixels whose neighborhoods
are not wholly contained in **G**.
w and h are the width and height, in pixels, defining a neighborhood.

```
procedure enhance_image(F,G,w,h);
{
for r := 0 to MaxRow - 1
   for c := 0 to MaxCol - 1
      {
      if [r,c] is a border pixel then G[r,c] := F[r,c];
      else G[r,c] := compute_using_neighbors ( F, r, c, w, h );
      } ;
}
procedure compute_using_neighbors ( IN, r, c, w, h )
{
using all pixels within w/2 and h/2 of pixel IN[r,c],
compute a value to return to represent IN[r,c]
}
```

Detecção de transições (*edges*)

- Operadores diferenciais de sinais 1D

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$



Máscara 1D centrada

mask M = [−1, 0, 1]

S_1			12	12	12	12	12	24	24	24	24	24
S_1	$\otimes$	M	0	0	0	0	12	12	0	0	0	0

(a) S_1 is an upward step edge

S_2			24	24	24	24	24	12	12	12	12	12
S_2	$\otimes$	M	0	0	0	0	-12	-12	0	0	0	0

(b) S_2 is a downward step edge

S_3			12	12	12	12	15	18	21	24	24	24
S_3	$\otimes$	M	0	0	0	3	6	6	6	3	0	0

(c) S_3 is an upward ramp

S_4			12	12	12	12	24	12	12	12	12	12
S_4	$\otimes$	M	0	0	0	12	0	-12	0	0	0	0

(d) S_4 is a bright impulse or "line"

Máscaras 1D

mask M = $[-1, 2, -1]$

S_1			12	12	12	12	12	24	24	24	24	24
S_1	$\otimes$	M	0	0	0	0	-12	12	0	0	0	0

(a) S_1 is an upward step edge

S_2			24	24	24	24	24	12	12	12	12	12
S_2	$\otimes$	M	0	0	0	0	12	-12	0	0	0	0

(b) S_2 is a downward step edge

S_3			12	12	12	12	15	18	21	24	24	24
S_3	$\otimes$	M	0	0	0	-3	0	0	0	3	0	0

(c) S_3 is an upward ramp

S_4			12	12	12	12	24	12	12	12	12	12
S_4	$\otimes$	M	0	0	0	-12	24	-12	0	0	0	0

(d) S_4 is a bright impulse or "line"

box smoothing mask M = $[1/3, 1/3, 1/3]$

S_1			12	12	12	12	12	24	24	24	24	24
S_1	$\otimes$	M	12	12	12	12	16	20	24	24	24	24

(a) S_1 is an upward step edge

S_4			12	12	12	12	24	12	12	12	12	12
S_4	$\otimes$	M	12	12	12	16	16	16	12	12	12	12

(d) S_4 is a bright impulse or "line"

Gaussian smoothing mask M = $[1/4, 1/2, 1/4]$

S_1			12	12	12	12	12	24	24	24	24	24
S_1	$\otimes$	M	12	12	12	12	15	21	24	24	24	24

(a) S_1 is an upward step edge

S_4			12	12	12	12	24	12	12	12	12	12
S_4	$\otimes$	M	12	12	12	15	18	15	12	12	12	12

Propriedades

- Operadores diferenciais
  - as coordenadas das máscaras tem sinais opostos para que se obtenha uma resposta máxima quando existem transições de intensidade (contraste)
  - A soma dos valores é zero para que a resposta seja zero quando a região é constante
  - As máscaras de primeira derivada produzem valores absolutos elevados em pontos de grande contraste
  - As máscaras de segunda derivada produzem cruzamentos por zero em pontos de grande contraste
- Operadores de suavização
  - os elementos da máscara são positivos e somam um, de modo a que a saída é igual à entrada em regiões de constante intensidade
  - A quantidade de suavização e remoção de ruído é proporcional à dimensão da máscara
  - Transições abruptas (step edges) são tanto mais espalhadas (blurred) quanto maior for a dimensão da máscara

- Gradiente duma função

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

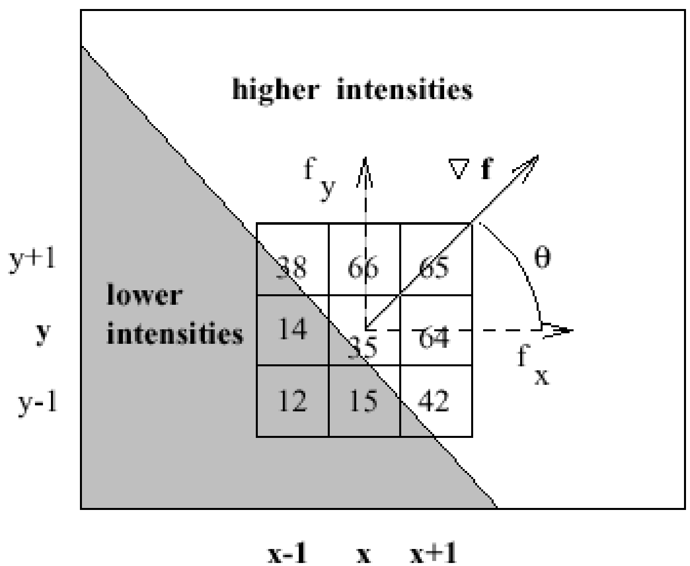
$$\frac{\partial f}{\partial x} \equiv f_x \approx \frac{1}{3}[\left(I(x+1,y) - I(x-1,y)\right)/2 + \left(I(x+1,y-1) - I(x-1,y-1)\right)/2 + \left(I(x+1,y+1) - I(x-1,y+1)\right)/2]$$

$$|\nabla f| \approx \sqrt{f_x^2 + f_y^2}$$

$$\theta \approx \tan^{-1}(f_y / f_x)$$

$$\frac{\partial f}{\partial y} \equiv f_y \approx \frac{1}{3}[\left(I(x,y+1) - I(x,y-1)\right)/2 + \left(I(x-1,y+1) - I(x-1,y-1)\right)/2 + \left(I(x+1,y+1) - I(x+1,y-1)\right)/2]$$

Exemplo:



f_y = ((38-12)/2 + (66-15)/2 + (65-42)/2) / 3
= (13 + 25 + 11) / 3 = 16

f_x = ((65-38)/2 + (64-14)/2 + (42-12)/2) / 3
= (13 + 25 + 15) / 3 = 18

$\theta = \tan^{-1}(16 / 18) = 0.727$ rad
= 42 degrees

$|\nabla f| = (16^2 + 18^2)^{1/2} = 24$

Frequentemente ignora-se a divisão por 6 (constante que será a mesma para toda a imagem) para reduzir os cálculos, resultando em estimativas escaladas (6 vezes superiores) do gradiente

Detectores de pontos de contorno

Qual a vizinhança?

Prewitt: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$

Sobel: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$

Roberts: $M_x = \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$

Calcula o gradiente da vizinhança, não no pixel central

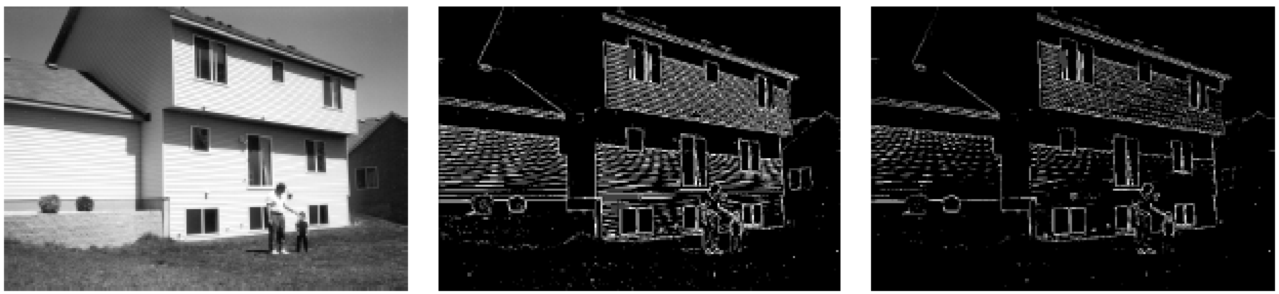
b) top 5% da soma dos valores absolutos das respostas às duas máscaras de Roberts

c) Top 5% v.q.m. das respostas às duas máscaras de Roberts

d) Top 2% da soma do valor absoluto das respostas â máscara na diração y $[-1, +1]^t$

e) Top 3% da soma do valor absoluto das respostas â máscara na diração x [-1, +1]

f) (d) OR (e)



(a) (b) (c)

(d) (e) (f)

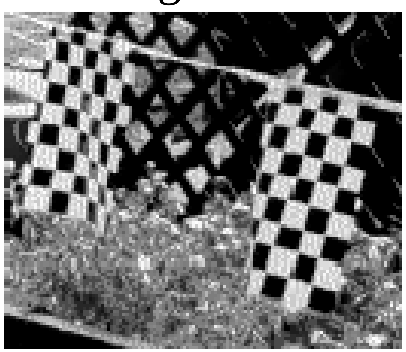
O detector de contornos de canny é muito popular e eficiente. Este algoritmo detecta os contornos utilizando pesquisa de máximos locais no valor do gradiente.

Detector de contornos de Canny


Original


Canny ($\sigma = 1$)


Canny ($\sigma = 4$)


Original


Canny ($\sigma = 1$)

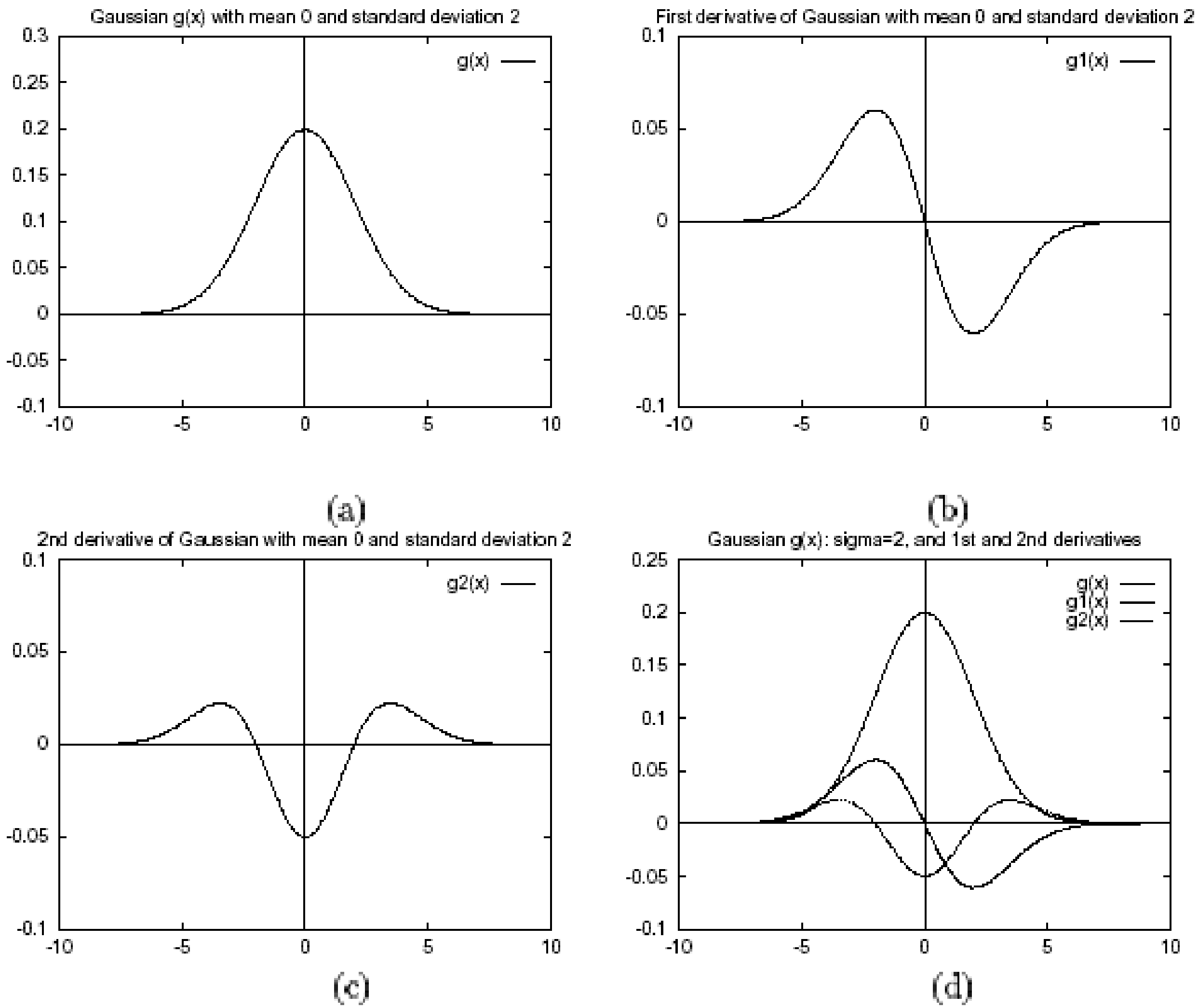

Roberts (20%)

Compute thin connected edge segments in the input image.

$\mathbf{I}[\mathbf{x},\mathbf{y}]$: input intensity image; σ : spread used in Gaussian smoothing;
$\mathbf{E}[\mathbf{x},\mathbf{y}]$: output binary image;
$\mathbf{IS}[\mathbf{x},\mathbf{y}]$: smoothed intensity image;
$\mathbf{Mag}[\mathbf{x},\mathbf{y}]$: gradient magnitude; $\mathbf{Dir}[\mathbf{x},\mathbf{y}]$: gradient direction;
T_{low} is low intensity threshold; T_{high} is high intensity threshold;

procedure Canny($\mathbf{I}[], \sigma$);
{
    $\mathbf{IS}[]$ = image $\mathbf{I}[]$ smoothed by convolution with Gaussian $G_\sigma(x,y)$;
    use Roberts operator to compute $\mathbf{Mag}[x,y]$ and $\mathbf{Dir}[x,y]$ from $\mathbf{IS}[]$;
    Suppress_Nonmaxima($\mathbf{Mag}[], \mathbf{Dir}[], T_{low}, T_{high}$);
    Edge_Detect($\mathbf{Mag}[], T_{low}, T_{high}, \mathbf{E}[]$);
}
procedure Suppress_Nonmaxima($\mathbf{Mag}[], \mathbf{Dir}[]$);
{
    define +Del[4] = (1,0), (1,1), (0,1) (-1,1);
    define -Del[4] = (-1,0), (-1-,1), (0,-1) (1,-1);
      for x := 0 to MaxX-1;
      for y := 0 to MaxY-1;
      {
        direction := ($\mathbf{Dir}[x,y] + \pi/8$) modulo $\pi/4$;
        if ($\mathbf{Mag}[x,y] \leq \mathbf{Mag}[(x,y) + Del[direction]]$) **then** $\mathbf{Mag}[x,y] := 0$;
        if ($\mathbf{Mag}[x,y] \leq \mathbf{Mag}[(x,y) + -Del[direction]]$) **then** $\mathbf{Mag}[x,y] := 0$;
      }
} **procedure** Edge_Detect($\mathbf{Mag}[], T_{low}, T_{high}, \mathbf{E}[]$);
{
    for x := 0 to MaxX - 1;
    for y := 0 to MaxY - 1;
      {
        if ($\mathbf{Mag}[x,y] \geq T_{high}$) **then** Follow_Edge($x, y, \mathbf{Mag}[], T_{low}, T_{high}, \mathbf{E}[]$);
      } ;
}
procedure Follow_Edge($x, y, \mathbf{Mag}[], T_{low}, T_{high}, \mathbf{E}[]$);
{
    $\mathbf{E}[x,y] := 1$;
    while $\mathbf{Mag}[u,v] > T_{low}$ for some 8-neighbor $[u,v]$ of $[x,y]$
      {
        $\mathbf{E}[u,v] := 1$;
        $[x,y] := [u,v]$;
      } ;
}

Algorithm 24: Canny Edge Detector



Caso 1D:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$g'(x) = \frac{-x}{\sigma^2} g(x)$$

$$g''(x) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right) g(x)$$

Caso 2D:

$$h(x, y) = g(r)$$

$$r = \sqrt{x^2 + y^2}$$

Filtro Gaussiano

c = 90: constante multiplicada a todos os elementos do filtro tal que o menor valor do filtro é 1

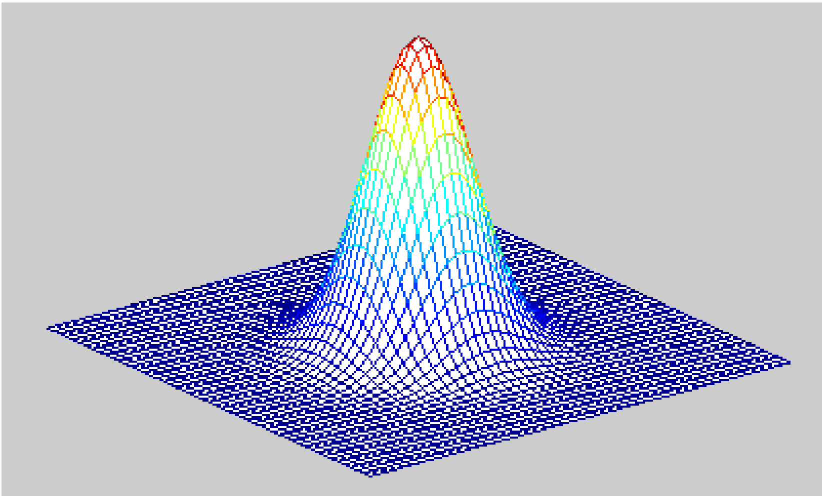
$$\mathbf{G}_{3\times3} = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}; \qquad \mathbf{G}_{7\times7} = \begin{array}{|c|c|c|c|c|c|c|}\hline 1 & 3 & 7 & 9 & 7 & 3 & 1 \\ \hline 3 & 12 & 26 & 33 & 26 & 12 & 3 \\ \hline 7 & 26 & 55 & 70 & 55 & 26 & 7 \\ \hline 9 & 33 & 70 & 90 & 70 & 33 & 9 \\ \hline 7 & 26 & 55 & 70 & 55 & 26 & 7 \\ \hline 3 & 12 & 26 & 33 & 26 & 12 & 3 \\ \hline 1 & 3 & 7 & 9 & 7 & 3 & 1 \\ \hline \end{array}$$

Figure 5.20: (Left) A 3×3 mask approximating a Gaussian obtained by matrix multiplication $[1,2,1]^t \otimes [1,2,1]$; (Right) a 7×7 mask approximating a Gaussian with $\sigma^2 = 2$ obtained by using Equation 5.16 to generate function values for integers x and y and then setting $c = 90$ so that the smallest mask element is 1.

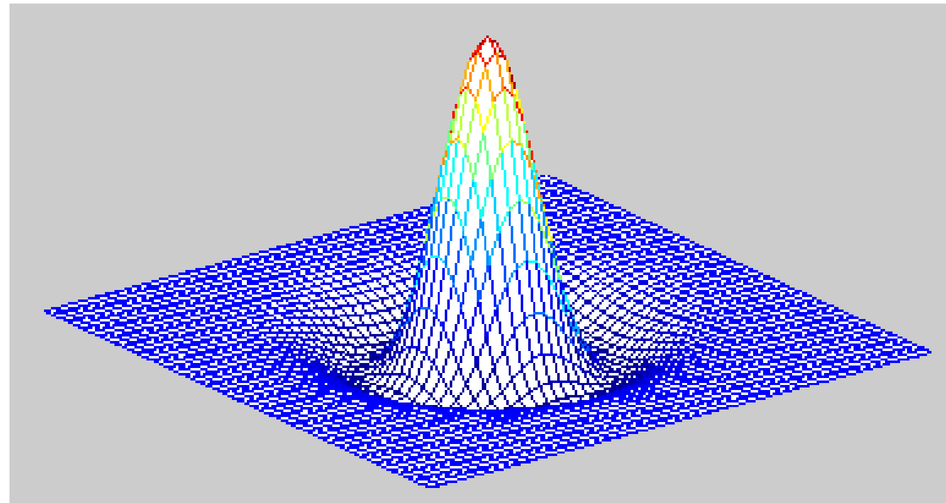
Some Useful Properties of Gaussians

1. Weight decreases smoothly to zero with distance from the origin, meaning that image values nearer the central location are more important than values that are more remote; moreover, the spread parameter σ determines how broad or focused the neighborhood will be. 95% of the total weight will be contained within 2σ of the center.
2. Symmetry about the abscissa; flipping the function for convolution produces the same kernel.
3. Fourier transformation into the frequency domain produces another Gaussian form, which means convolution with a Gaussian mask in the spatial domain reduces high frequency image trends smoothly as spatial frequency increases.
4. The second derivative of a 1D Gaussian $g''(x)$ has a smooth center lobe of negative area and two smooth side lobes of positive area: the zero crossings are located at $-\sigma$ and $+\sigma$, corresponding to the inflection points of $g(x)$ and the extreme points of $g'(x)$.
5. A second derivative filter based on the Laplacian of the Gaussian is called a **LOG filter**. A LOG filter can be approximated nicely by taking the difference of two Gaussians $g''(x) \approx c_1 e^{-\frac{x^2}{2\sigma_1^2}} - c_2 e^{-\frac{x^2}{2\sigma_2^2}}$, which is often called a **DOG** filter (for **D**ifference **O**f **G**aussians). For a positive center lobe, we must have $\sigma_1 < \sigma_2$; also, σ_2 must be carefully related to σ_1 to obtain the correct location of zero crossings and so that the total negative weight balances the total positive weight.
6. The LOG filter responds well to intensity differences of two kinds – small blobs coinciding with the center lobe, and large step edges very close to the center lobe.

$g(x,y)$



$$L(x,y)=\frac{\partial^2 g(x,y)}{\partial x^2}+\frac{\partial^2 g(x,y)}{\partial y^2}$$



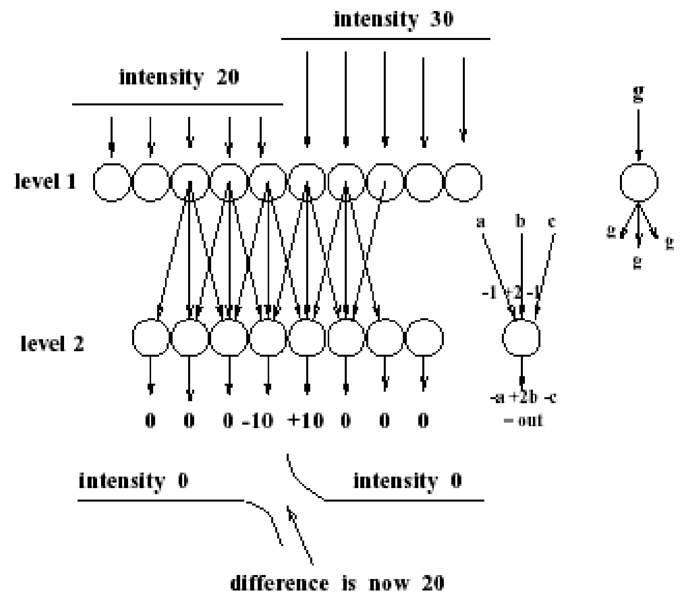
$-L(x,y) \longrightarrow$ Chapéu mexicano (*sombrero*)

0	0	0	-1	-1	-2	-1	-1	0	0	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	-2	-7	-15	-22	-23	-22	15	-7	-2	0
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-2	-9	-23	-1	103	178	103	-1	-23	-9	-2
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
0	-2	-7	-15	-22	-23	-22	15	-7	-2	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	0	0	-1	-1	-2	-1	-1	0	0	0

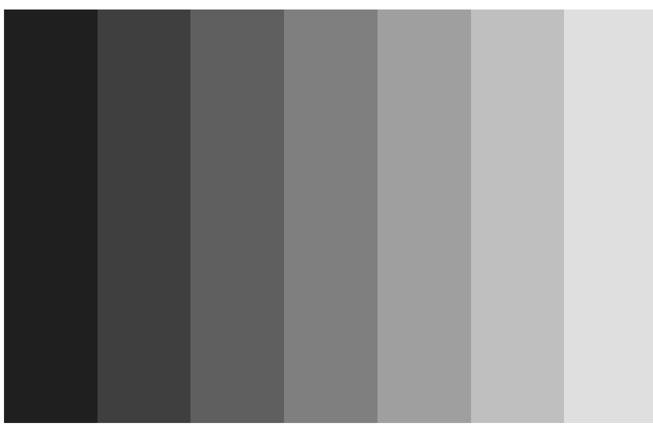
$\longleftarrow$ Máscara 11x11 (σ^2=2)

O Psicólogo Mach reparou que os humanos percecionam contornos entre duas regiões como se o contorno fosse separado da imagem para exagerar as diferenças de intensidade

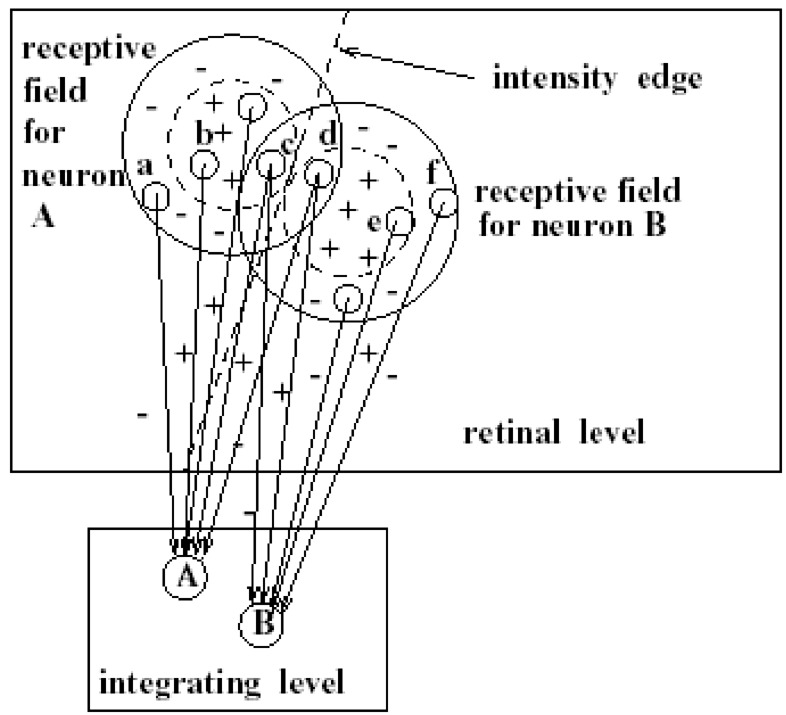
Modelo neuronal e o efeito das bandas de Mach



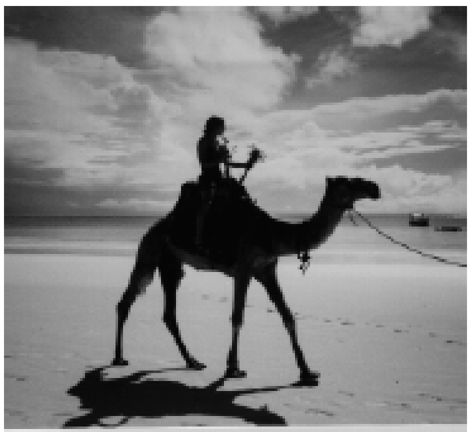
- Células da retina (nível 1) são sensíveis à intensidade luminosa
- Células de integração (nível 2) são sensíveis às transições de intensidade



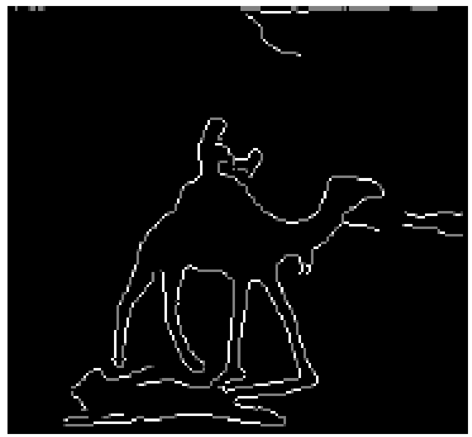
Bandas de Mach



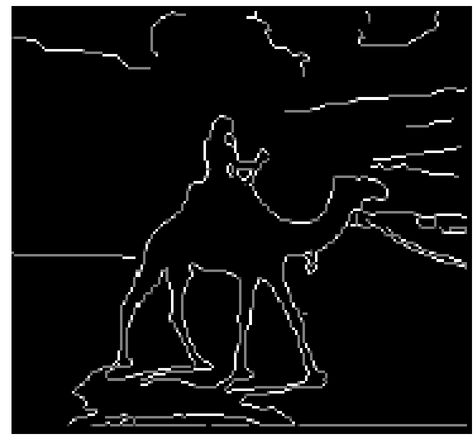
- Filtro LOG ajuda a explicar o SVH (baixo nível)
  - Objectivo primeiro é a construção do esboço fundamental (*primal sketch: lines, edges, blobs*)
- Análise multiresolução
  - filtragem LOG, com elevado σ, permite a detecção das estruturas principais existentes na imagem, enquanto que os detalhes se obtêm fazendo o processamento com σ pequeno.



original



smoothed $\sigma = 4$

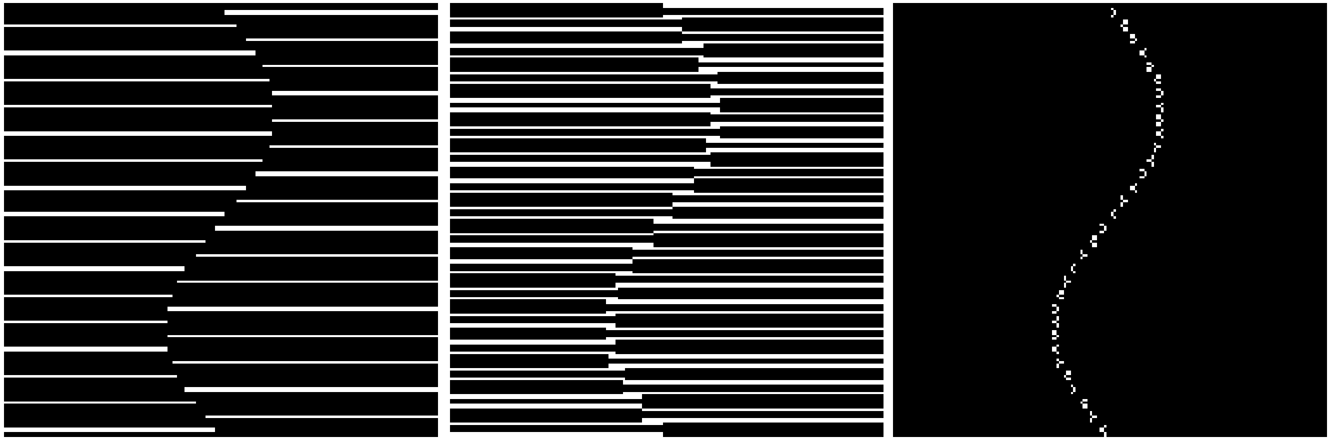


smoothed $\sigma = 1$

Agrupamento perceptual - linhas virtuais

Linha virtual

Saídas de dois filtros LOG



Sigmas altos detectam contornos fortes e sigmas mais baixos detectam mais detalhe.

A fusão dos níveis é realizada em níveis superiores da perceção

Linhas virtuais em imagens reais